

# K<sub>E</sub>T<sub>C</sub>indy で作成した PDF スライド教材 による授業実践について

木更津工業高等専門学校・基礎学系 山下 哲 (Satoshi Yamashita)

Division of Natural Science,  
National Institute of Technology, Kisarazu College

長野工業高等専門学校・一般科 小林 茂樹 (Shigeki Kobayashi)

General Education,  
National Institute of Technology, Nagano College

芝浦工業大学・工学部 牧下 英世 (Hideyo Makishita)

College of Engineering,  
Shibaura Institute of Technology

東邦大学・理学部 高遠 節夫 (Setsuo Takato)

Faculty of Science,  
Toho University

## 1 K<sub>E</sub>T<sub>C</sub>indy による PDF スライド教材の作成

### 1.1 K<sub>E</sub>T<sub>C</sub>indy とは

L<sup>A</sup>T<sub>E</sub>X 文書へ図表を簡単に挿入するために、2006年から、数学ソフト (Maple, Mathematica など) のマクロパッケージとして K<sub>E</sub>T<sub>p</sub>ic を開発し、2010年に数値計算ソフト Scilab 版 K<sub>E</sub>T<sub>p</sub>ic として完成した。2014年には、動的幾何ソフト Cinderella と連携して K<sub>E</sub>T<sub>C</sub>indy に進化し、GUI (Graphic User Interface) も装備された。K<sub>E</sub>T<sub>C</sub>indy のインストール方法については、web ページ

<https://www65.atwiki.jp/ketcindy/>

の「K<sub>E</sub>T<sub>C</sub>indy の準備 (インストール方法)」を参照せよ。現在、K<sub>E</sub>T<sub>C</sub>indy では、Scilab 以外の数学ソフト (maxima, R, Risa/Asir, FriCAS) も援用できるよう改良されており、K<sub>E</sub>T<sub>C</sub>indy で作成できる作業を表 1 にまとめる。また、K<sub>E</sub>T<sub>C</sub>indy をインストールすると

`/Users/(User's Name)/ketcindy/ketsample/samples/`

に表 1 のサンプルフォルダ群があり、これらの作業方法の参考になる。

本論文では、K<sub>E</sub>T<sub>C</sub>indy による PDF スライド教材の作成方法について解説し、実際、この PDF スライド教材を用いたアクティブ・ラーニング型授業について報告する。

表 1. KpTCindy のサンプルフォルダー一覧

s01	Geometric Figure	s09	Surface
s02	Graph of Function	s10	Calling Maxima
s03	Making Table	s11	Calling Asir
s04	Bćzier Curve	s12	Calling Fricas
s05	3D Figure	s13	Calling Mesthlab
s06	Animation	s14	Data Processing
s07	Slide for Presentation	s15	C
s08	Calling R		

## 1.2 beamer パッケージについて

$\LaTeX$  ユーザーの間では、PDF スライドを作成するツールとして beamer パッケージがよく利用されている。本節では、beamer パッケージについて簡単に解説する。beamer パッケージの特徴は、 $\LaTeX$  コマンドとスライド用コマンドを  $\LaTeX$  ドキュメント上で併用できることにある。1 ページのスライドは frame 環境で以下のように作成する。

```
\documentclass[dvipdfm]{beamer}
\begin{document}
  \begin{frame}{フレームのタイトル}
    \frametitle
    テキストを記入
  \end{frame}
\end{document}
```

frame 環境を繰り返すことにより、複数ページのスライドが作成できる。

- block 強調表示

囲みで強調したい場合、以下のように使用する。

```
\begin{block}{ブロックのタイトル}
  ブロックのテキストを記入
\end{block}
```

- 定義・定理表示

定義や定理などを囲みで強調したい場合、以下のように使用する。

```
\begin{definition}{[\cite{**}]}
  定義のテキストを記入
\end{definition}
```

- **overlay 機能**

overlay (覆うもの) は動的なコマンドで、テキストを覆い隠しておいて、順に1つずつ出力していく。順番を管理する内部変数は `beamerpauses` で、ページ送り・戻りでこの数値が増減する。例えば、コマンド `\onslide` は、テキストを順に表示できる。また、`itemize` 環境で各 `\item` の最後にコマンド `\pause` を入れると、箇条書きが順に出力される。

```
\begin{itemize}
\item
  箇条書き 1 のテキストを記入。 \pause
\item
  箇条書き 2 のテキストを記入。
\end{itemize}
```

- **link 制御と button 表示**

`hyperlink` を制御するために、以下のようにボタンを利用できる。

```
\hyperlink<overlay specification>{ターゲット}{
  テキスト
}{target name}{\beamerbutton{ターゲットへ}}
```

次のスライドへ進む場合：`\hyperlinkslidenext{\beamergetobutton{}}`

前のスライドへ戻る場合：`\hyperlinkslideprev{\beamerreturnbutton{}}`

スライドを再表示させる場合：`\againframe{target name}`

これらの機能を利用して、動的な PDF スライドを作成できる。

### 1.3 PDF スライドの作成方法について

前節で解説したように、`beamer` パッケージを利用する場合、挿入する図表は別の方法 (`Tikz` や画像ファイルなど) で作成しなければならない。`KETCindy` では、図表とともにスライドを作成できるようにした。本節では、`KETCindy` による PDF スライドの作成方法について、表 1 のサンプルフォルダ `s07` を使って簡単に解説する。本論文では、`macintosh` の場合で解説するが、`windows` の場合でも同様にできる。

まず、デスクトップ上に作業用フォルダ `mywork` を作成する。サンプルフォルダ

```
/Users/(User's Name)/ketcindy/ketsample/samples/s07slides/
```

を開いて、2つのファイル

`s0701basic.cdy` : Cinderella 実行ファイル

`s0701basic.txt` : スライド原稿用テキストファイル

を `mywork` フォルダにコピーする。PDF スライドを作成するには、スライド原稿用テキストファイルにスライドの原稿を作成し、Cinderella 実行ファイルで PDF スライドを生成させる。また、Cinderella 実行ファイルでは、スライドに挿入する図やパラパラ動画なども作成できる。

次に、これら2つのファイルを使って、実際に PDF スライドを作成してみよう。スライド原稿用テキストファイル `s0701basic.txt` を開くと、以下のプログラムが書かれている。

```

1 タイトル::slide0//
2 %%タイトル::slide0::kabegami//
3 %%title:://
4
5 %%%%%%%%%%%//
6 main::メインスライド1//
7
8 %%%%%%%%%%%//
9 new::itemize 環境の使い方//
10
11 itemize//
12 item::内容//
13 item::結論//
14 end//
15
16 %%%%%%%%%%%//
17 new::enumerate 環境の使い方//
18
19 enumerate::[(1)]//
20 item::番号を変えるには、例えば [(1)]//
21 item::後は同じ//
22 end//

```

`s0701basic.cdy` を実行すると、上記プログラムが読み込まれ、自動的に、スライド用  $\LaTeX$  文書ファイル `s0701basic.tex` が生成、コンパイルされ、PDF スライドが出力される。では、このプログラムについて解説しよう。

1行目のプログラムにより、タイトルスライドが表示される。各行の最後にある//により、自動生成される `s0701basic.tex` で改行される。2行目と3行目は、行頭%%により、`s0701basic.tex` でコメントアウトされる。タイトルスライドは、Cinderella 実行ファイル `s0701basic.cdy` で作成され、`fig` フォルダ (`s0701basic.cdy` を実行すると自動的に生成される) 内に  $\LaTeX$  ソースファイル `slide0.tex` として保存される。5行目


はスライドのページ区切りを表している。6行目は、ページの中央に「メインスライド 1」と表示させる。8行目と17行目は、新しいページの上方に「itemize 環境の使い方」と表示させる。10行目から14行目までは、itemize 環境で箇条書きを表示させる。19行目から22行目までは、enumerate 環境で番号付き箇条書きを表示させる。

さらに、s0701basic.cdyを開き、上部ツールバーから「スクリプト」>「CindyScript」を選択しクリックすると、スクリプトエディタが開く。スクリプトエディタには、以下のプログラムが書かれている。

```

1  Ketinit();
2
3  Slidework(Dircdy);
4
5  //Addpackage(["pict2e"]);
6  //Addpackage(["[dvipdfmx]{media9}", "[dvipdfmx]{animate}", "ketmedia"]);
7
8  Setslidebody("black");
9  Setslidehyper("");
10
11 Texcom("\Large\bf\boldmath");
12 Fontsize("s");
13
14 Settitle([
15  "s{60}{20}{メインタイトル}",
16  "s{60}{50}{名前}",
17  "s{60}{60}{所属}",
18  "s{60}{70}{情報}"
19  ], ["Color=[1,1,0,0]"]);
20
21 Windispg();

```

スクリプトエディタ右上端のギアマークをクリックすると、スクリプトエディタに書かれたプログラムが実行され、mywork フォルダ内に K<sub>E</sub>T Cindy 用一括処理ファイル kc.sh と fig フォルダが出現する。このとき、fig フォルダ内にも kc.sh が生成される。では、このプログラムについて解説しよう。

1行目で K<sub>E</sub>T Cindy を初期化し、21行目までのプログラムを実行し、Cinderella メイン画面に表示させる。3行目はスライド関係ファイルの作成場所を指定し、Dircdy で s0701basic.cdy があるフォルダを指定できる。5行目と6行目は L<sup>A</sup>T<sub>E</sub>X で使用する sty ファイルを指定し、文頭の//でコメントアウトされている（実行しない）。11行目は s0701basic.tex に挿入したい文を指定し、12行目は作成する図に挿入する文字のフォントサイズを \small と指定する。14行目から19行目まででタイトルページに書き込む情報を指定する。s{60}{20}は配置する場所を指定しており、横60mm、縦20mmの位置の南側に書き込むことを意味する。

タイトルスライドを作成するには、Cinderellaメイン画面で「Title」ボタンをクリックすればよい。すると、タイトルスライドが `slide0.tex` として `fig` フォルダ内に生成され、PDF で表示される。さらに、「Slide」ボタンをクリックすると、`mywork` フォルダ内に `s0701basic.tex` が生成され、PDF で表示される。

基本的な操作は以上である。この他、`beamer` と同様、`overlay` 機能も可能である。具体的には、`s07slides` フォルダ内の `s0703repeat.txt` を開くと、以下のプログラムが書かれている。

```
1 new::enumerate 環境の使い方//
2 %repeat=3//
3
4 enumerate::[(1)]//
5 %thin[2,-]::item::番号を変えるには、例えば [(1)]//
6 %thin[3,-]::item::後は同じ//
7 end//
```

2行目は、同じスライドを3ページ繰り返すことを指定する。5行目は、`%thin[2,-]::` で2ページ以降に表示させることを指定し、それ以外(1ページ目)のページでは薄文字で表示させることを指定する。6行目も同様である。また、パラパラ動画を作成し、スライドに表示させるには、`s07slides` フォルダ内の `s0705para.cdy` と `s0705para.txt` を参照せよ。

## 2 PDFスライド教材を用いたアクティブ・ラーニング型授業

カレッジ級数学とは、高校1年次から理工系大学2年次まで(高専では1年次から5年次までに相当)に修得すべき数学のことであり、数学の理論体系に関する記号的理解(記号体系と概念構造を同化する能力)や論理的理解が要求される[1]。これらの要求を学生に実現させるには、理論の基礎事項を順次関連付ける必要があり、理工系の学生でさえ視覚化されたイメージ付けを行わないと認知的つながりを構築しにくい。このため、カレッジ級数学の授業では、図入りプリントや図入りスライドを利用することで、学生の理解を効果的に促進できる。

カレッジ級数学の授業では、講義だけでなく演習も取り入れて、学生の数学的理解の促進を図っている。そこで、学生が自主的に学習し修得できるような授業スタイルとして、アクティブ・ラーニング型授業が盛んに行われるようになってきた。アクティブ・ラーニング型授業では、講義時間を抑え、演習時間を多く確保することにより、学生の主体的な学びを促進する。そこで、2017年度前期に、高専2年生(高校2年生相当)を対象として、平面ベクトル・空間ベクトルの内容をアクティブ・ラーニング型授業で学ばせた。毎回の授業方法(90分)は以下の通りである。

- 1) 前回の授業で回収した課題や小テストを返却し、注意点を指摘する。次回使用する課題を配付する。(5分)

- 2) 予習確認テストと公式テストを実施し、学生同士で相互採点する。(10分)
- 3) PDF スライド教材を用いて、講義する。(15分)
- 4) 学生に課題を完成させ、回収する。(50分)
- 5) 前回の授業で回収した課題の復習確認テストを実施する。ただし、回収後教員が採点し、次回の授業で返却する。(10分)

この授業を実施する目的は、学生に予習・復習することで理解しやすくなり、課題の時間に学生同士で教え合うことで理解が深められることを体験させることにある。

最後の授業で学生の授業アンケートを実施した。アンケート項目は以下の通りである。

- (1) 毎回の授業で、あなたはどの程度予習しますか。  
① 0分 ② 30分 ③ 1時間 ④ 1時間30分 ⑤ 2時間以上
- (2) 毎回の授業の予習で、あなたはどの程度課題を仕上げられますか。  
① 0% ② 30% ③ 50% ④ 80% ⑤ 100%
- (3) 毎回の授業で、あなたは教科書の内容をどの程度理解できますか。  
① 0% ② 30% ③ 50% ④ 80% ⑤ 100%
- (4) 毎回の授業の講義で、あなたは講義内容をどの程度理解できますか。  
① 0% ② 30% ③ 50% ④ 80% ⑤ 100%
- (5) 毎回の授業の講義で、あなたはスライドと板書のどちらがよいですか。  
① スライド ② どちらでもよい ③ 板書
- (6) 毎回の授業の演習(『学び合い』)で、あなたはどの程度課題を仕上げられますか。  
① 0% ② 30% ③ 50% ④ 80% ⑤ 100%
- (7) 毎回の授業の予習確認テストは、あなたにとってどの程度難しいですか。  
① 簡単 ② 少し簡単 ③ 普通 ④ 少し難しい ⑤ 難しい
- (8) 毎回の授業の公式テストは、あなたにとってどの程度難しいですか。  
① 簡単 ② 少し簡単 ③ 普通 ④ 少し難しい ⑤ 難しい
- (9) 毎回の授業で、あなたはどの程度復習しますか。  
① 0分 ② 30分 ③ 1時間 ④ 1時間30分 ⑤ 2時間以上
- (10) 毎回の授業の復習確認テストは、あなたにとってどの程度難しいですか。  
① 簡単 ② 少し簡単 ③ 普通 ④ 少し難しい ⑤ 難しい

このアンケート結果は以下のようになった。

- (1) 毎回の授業で、予習時間は30分～1時間程度である。

- (2) 毎回の授業の予習で，課題を 30 % または 80 % 仕上げられる。
- (3) 毎回の授業で，教科書の内容を約 50 % 理解できる。
- (4) 毎回の授業の講義で，講義内容を約 50 % 理解できる。
- (5) 毎回の授業の講義で，スライドと板書のどちらでもよい。
- (6) 毎回の授業の演習（『学び合い』）で，課題を 50 % または 100 % 仕上げられる。
- (7) 毎回の授業の予習確認テストは，少し難しい。
- (8) 毎回の授業の公式テストは，簡単または少し難しい。
- (9) 毎回の授業で，復習時間は約 30 分である。
- (10) 毎回の授業の復習確認テストは，難しい。

### 3 まとめと今後の課題

KETCindyにより，図入りPDFスライド教材を容易に作成できるようになった。beamerパッケージと同様なoverlay機能だけでなく，パラパラ動画やアニメーションも挿入できる。このPDFスライド教材を利用することで，講義を理解しやすくし，講義時間も短縮できるので，アクティブ・ラーニング型授業に適している。学生による授業アンケート結果でも，(5)で板書とスライドに大きな違いがないこと，(2)と(6)を比較すると，アクティブ・ラーニング型授業により，課題の完成度が上がっていることが挙げられる。最後に，今後，以下の課題に取り組んでいく。

- 音声付き動画を挿入することで，予習用動画教材を作成する。
- KETCindyのスライド作成機能をさらに使いやすくなるよう改良する。

#### 謝辞

本研究は，京都大学数理解析研究所共同事業「数学ソフトウェアとその効果的教育利用に関する研究」による成果である。本研究はJSPS科研費16K01152の助成を受けている。

### 参考文献

- [1] 山下哲，「KETCindyによる図入りPDF教材の作成」，京都大学数理解析研究所講究録2011，pp.59-64，2017。